REMARKS

Claims 1-36 are pending in the application, of which Claims 1, 6, 10, 15, 19, 24, 28 and 33 are independent claims. Claims 1, 4-7, 10, 13, 14-16, 19, 22 and 23-25 have been rejected under 35 U.S.C. § 103(a) as being deemed unpatentable over Zolnowsky (U.S. Patent No. 5,826,081) in view of Sullivan (U.S. Patent No. 5,438,680). Claims 2, 3, 8, 9, 11, 12, 17, 18, 20, 21 and 26-27 have been rejected under 35 U.S.C. § 103(a) as being deemed unpatentable (U.S. Patent No. 6,324,162) over Zolnowsky (U.S. Patent No. 5,826,081) in view of Sullivan (U.S. Patent No. 5,438,680) and further in view of Najork et al. (U.S. Patent No. 6,377,934). The rejections are traversed.

Before discussing the cited references, another brief review of the Applicant's disclosure may be helpful. The Applicant claims a method for processing computing tasks in a multithreaded computing environment. The multiple threads can run on a single processor or multiple processors. A plurality of worker threads are defined, each working thread capable of processing a task. (*See* Fig. 3, worker threads W in a worker thread pool 30, task queues Q in a queue space 40.) Application programs execute software instructions on a processor to perform work. These instructions are divided into discrete tasks for processing.

In a multithreaded computing environment, the tasks are assigned to multiple worker threads for processing. The threads perform the task and return the results to the application program. A task space is defined as a plurality of task queues. Each task queue is capable of queuing a plurality of tasks and associated with a respective worker thread. A task scheduler assigns a task amongst the task queues in an essentially random fashion.

Turning to the cited references, Zolnowsky discusses a method for scheduling threads in a multiprocessor system. Runnable threads to be executed by a processor are stored in a global high priority real time queue and in a dispatch queue associated with each processor. Each processor selects the highest priority runnable thread from one of the queues. (*See* Col. 7, lines 12-56; and Fig. 5.)

Sullivan discusses a method for scheduling processes in a multiprocessor system. (*See* Col. 3, line 64-Col 4, line 3; and Col. 6, lines 9-12.)

Najork discusses a method of queuing universal resource locators dependent on the host address. (*See* Col. 3, lines 55-67.)

Applicant's respectfully disagree with the Office's suggestion that Zolnowsky teaches the applicant's disclosed task queue. Zolonowsky discusses the queuing of threads for a processor, not tasks for a thread. The applicant's task queue is capable of queuing a plurality of tasks to be processed by threads. As claimed in each independent claim, each task queue is associated with a separate thread. (*See* Applicant's specification Page 5, lines 3-8, Page 6, lines 3-7 and Page 7, lines 22-24.) By associating each task queue with a separate thread, lock contention problems associated with a single shared task queue are diminished. Also, the waiting time of tasks is reduced because tasks can be distributed amongst the task queues. Thus, threads are kept busy.

In contrast, Zolnowsky discusses a queue of runnable threads (dispatch queue) assigned to a processor. Zolnowsky's dispatch queue does not teach or suggest the applicant's disclosed task queue assigned to a thread. Zolnowsky does not even discuss assignment of tasks to threads. Zolnowsky merely discusses assignment of runnable threads to processors that are queued on a high priority real time queue or a dispatch queue. Zolnowsky does not keep threads busy, and does not really keep processors busy. It just prioritizes threads so that highest priority threads are run first.

In contrast to the cited references, the Applicant claims a method for scheduling tasks in a multithreaded system which may have a single processor. The system includes a plurality of worker threads capable of processing a task. Neither Zolnowsky nor Sullivan even discuss a task queue or associating a task queue with a respective worker thread. Furthermore, neither even discuss a task queue for use in a multithreaded system.

Zolnowsky and Sullivan merely discuss scheduling threads or processes in a multiprocessor system. Najork does not cure the deficiencies in the combination of Zolnowsky and Sullivan because the combination does not teach or suggest a task queue or associating a task queue with a respective worker thread.

Even in combination, Zolnowsky and Sullivan do not suggest the Applicant's claimed invention for defining a plurality of task queues and associating each task queue with a worker thread.

Patentably distinguishing claim language of independent Claims 1, 6, 19 and 24 reads, in pertinent part:

> defining a plurality of task queues, each task queue capable of queuing a plurality of tasks;

associating each task queue with a respective worker thread;

In addition, patentably distinguishable claim language of independent Claims 10 and 15 reads, in pertinent part:

> a plurality of task queues, each task queue capable of queuing a plurality of tasks and each task queue associated with a respective worker thread;

Claims 2-5 are dependent on Claim 1; Claims 7-9 are dependent on Claim 6; Claims 11-14 are dependent on Claim 10; Claims 16-18 are dependent on Claim 15; Claims 20-23 are dependent on Claim 19; Claims 25-27 are dependent on Claim 24; Claims 29-32 are dependent on Claim 28; and Claims 34-36 are dependent on Claim 33 respectively and thus include this limitation over the prior art. Accordingly, the present invention as now claimed is not suggested by the cited art. Reconsideration of the rejections under 35 U.S.C. § 103(a) is respectively requested.

CONCLUSION

In view of the above remarks, it is believed that all claims (claims 1-36) are in condition for allowance, and it is respectfully requested that the application be passed to issue. If the Examiner feels that a telephone conference would expedite prosecution of this case, the Examiner is invited to call the undersigned attorney.

Respectfully submitted,

HAMILTON, BROOK, SMITH & REYNOLDS, P.C.

By ____________________
Caroline M. Fleming
Registration No. 45,566
Telephone: (978) 341-0036
Facsimile: (978) 341-0136

Concord, MA 01742-9133
Dated: 3/10/04